

Preserving Exchange Tasks (or Contacts or Calendars) from Archive or Deletion Policies

Tasks, Calendars, and Contacts may be archived or deleted based on their calculated retention date, see <https://learn.microsoft.com/en-us/exchange/security-and-compliance/messaging-records-management/retention-age?source=recommendations>

Unfortunately it's not always possible to create a policy on a folder that prevents archiving or deleting using retention policies or Outlook. So Microsoft has a crazy workaround for this (last tested May 2024).

Outlined steps below to set a retention policy on Tasks folder. First create the personal retention policy to stop archiving. We need to get the PR_ARCHIVE_TAG (step needs to be added).


1. Install EWS editor

<https://github.com/dseph/EwsEditor/releases>

EWSEditor - 1.24.0 Latest

You can now set the Authority and Scope in the log and other tenants.

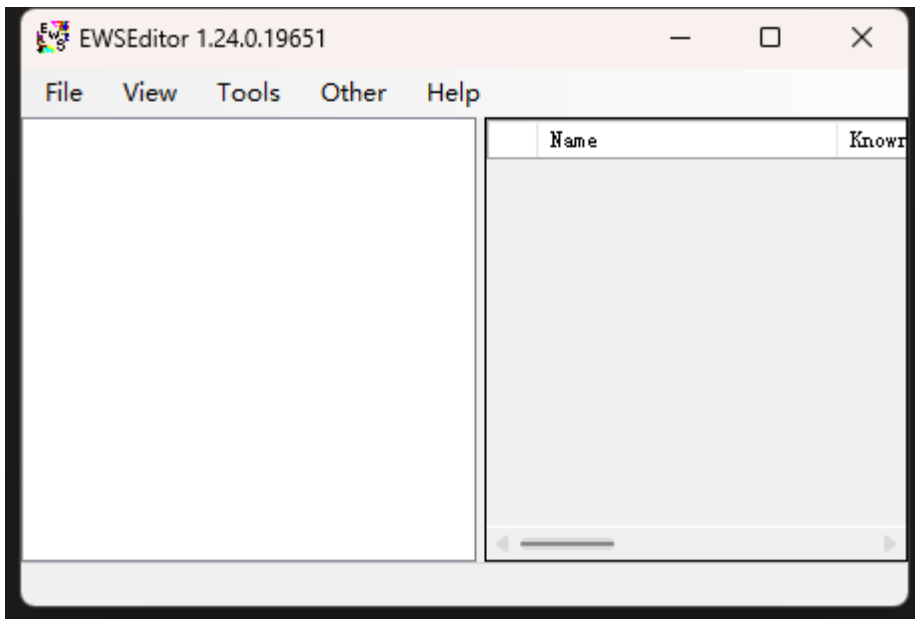
▼ Assets 3

 EwsEditor.1.24.-.bin.zip

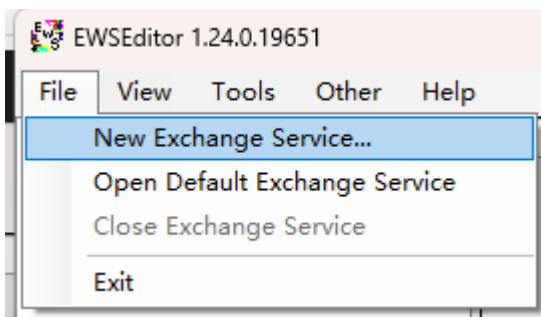
 [Source code \(zip\)](#)

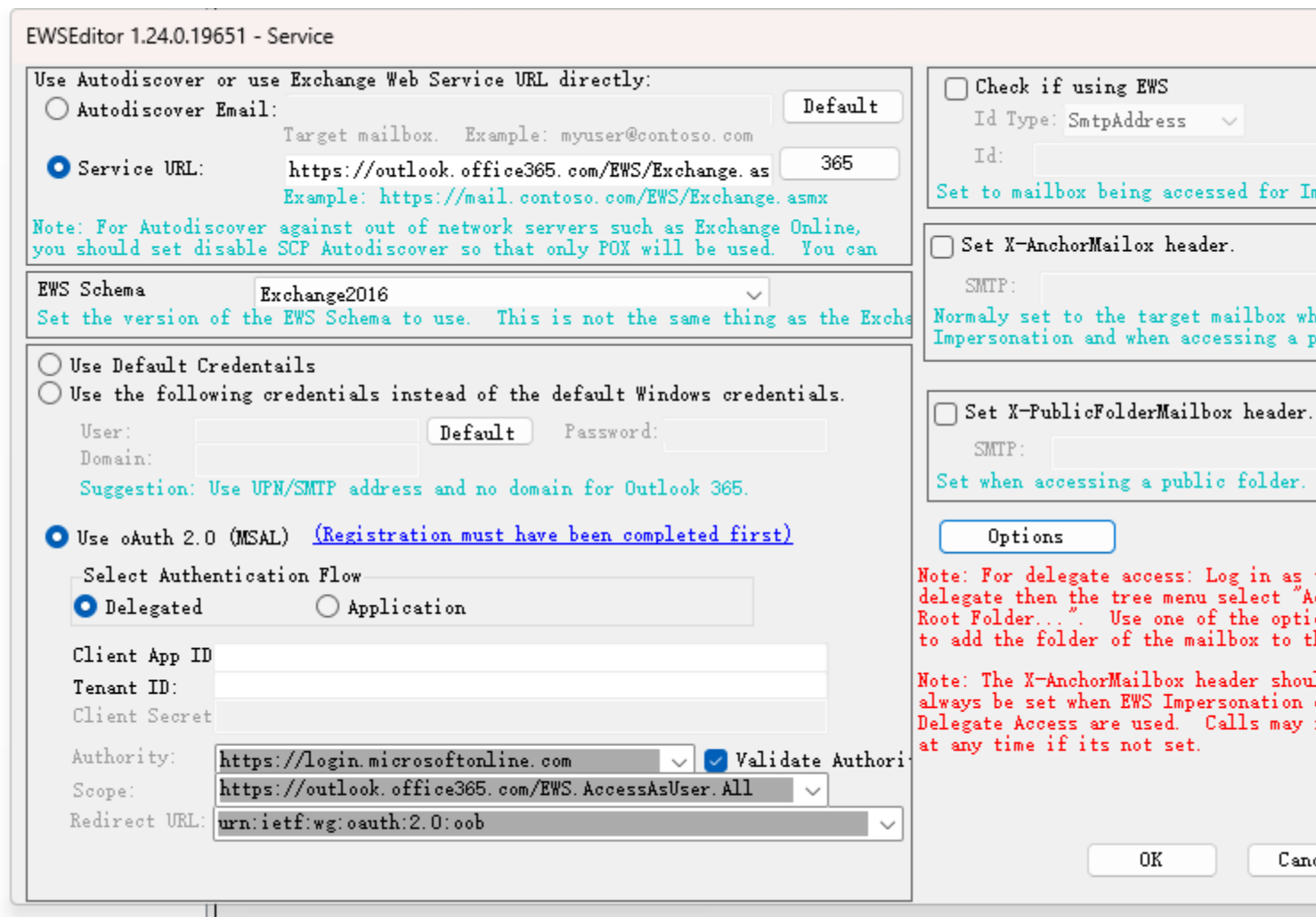
 [Source code \(tar.gz\)](#)

2. When you have it on your PC, it looks like this:



Then, click on the File and choose New Exchange Service:





3. login [App registrations - Microsoft Entra admin center](#)

entra.microsoft.com/#view/Microsoft_AAD_RegisteredApps/CreateApplicationBlac

Microsoft Entra admin center

Home > App registrations > Test_for_EWS > App registrations >

Register an application

* Name

The user-facing display name for this application (this can be changed later).

Test_for_EWS

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Contoso only - Single tenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Organizations)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Organizations, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. You can change this URI later, but a value is required for most authentication scenarios.

Public client/native (mobile ...

Then, we can use the App's client ID and your tenant ID to sign-in;

//Application ID is the client ID, and Directory ID is the tenant ID;

//please make sure the redirect URL is the one we just paste in the above picture;

^ Essentials

Display name : [Test for EWS](#)
Application (client) ID : cdbe888b-de5c-40de-a775-dcc0109d7080
Object ID : 736368c3-df4c-483f-b6a4-4b11c6d9eb03
Directory (tenant) ID : eb106505-c67c-4c49-9a1f-631dd4414ce4
Supported account types : [My organization only](#)

4.Add API permission

The screenshot shows the Microsoft Entra admin center interface. The left sidebar contains navigation options: Home, What's new, Diagnose & solve problems, Favorites, Identity, Overview, Users, Groups, Devices, Applications, Enterprise applications, App registrations, and Protection. The 'App registrations' section is highlighted. The main content area shows the configuration for 'test for EWS' under 'API permissions'. The 'API permissions' option is highlighted in the left-hand menu. The right-hand pane shows the 'Request API permissions' section, where 'office 365' is selected. A list of APIs is displayed, with 'Office 365 Exchange Online' highlighted. Below the list, the 'API / Permissions' section shows 'Microsoft Graph (User.Read)' and 'Office 365 Exchange Online'.

Request API permissions

[< All APIs](#)

03 Office 365 Exchange Online
<https://outlook.office.com>

What type of permissions does your application require?

Delegated permissions

Your application needs to access the API as the signed-in user.

Application permissions

Your application runs as a background service or daemon with signed-in user.

Permission	Admin consent required
> Calendars	
> Contacts	
> EAS	
▼ EWS (1)	
<input checked="" type="checkbox"/> EWS.AccessAsUser.All ⓘ Access mailboxes as the signed-in user via Exchange Web Services	No
> Exchange	

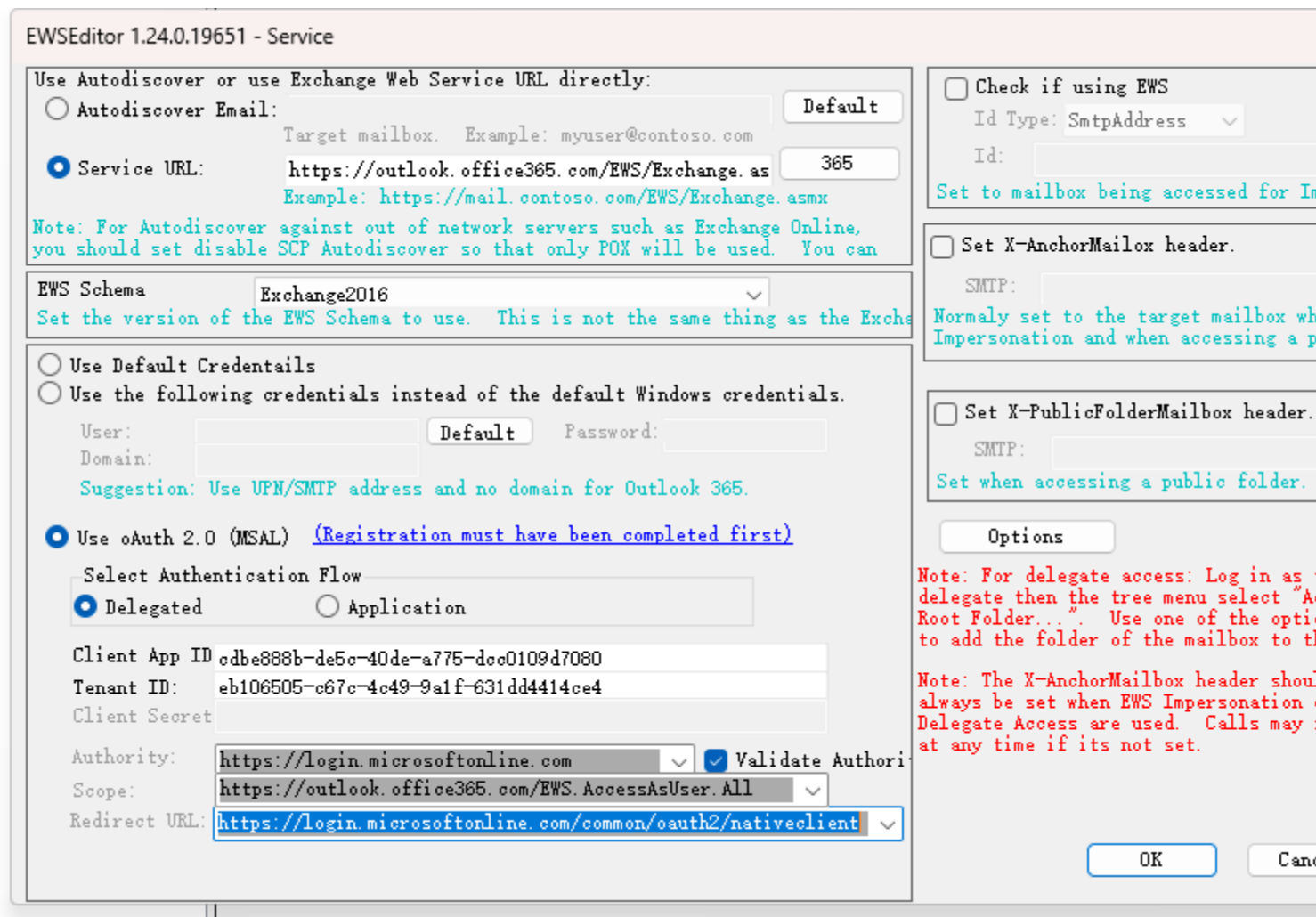
- Applications ^
 - Enterprise applications
 - App registrations
- Protection ▼
- Identity governance ▼

- API permissions**
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest
- Support + Troubleshooting

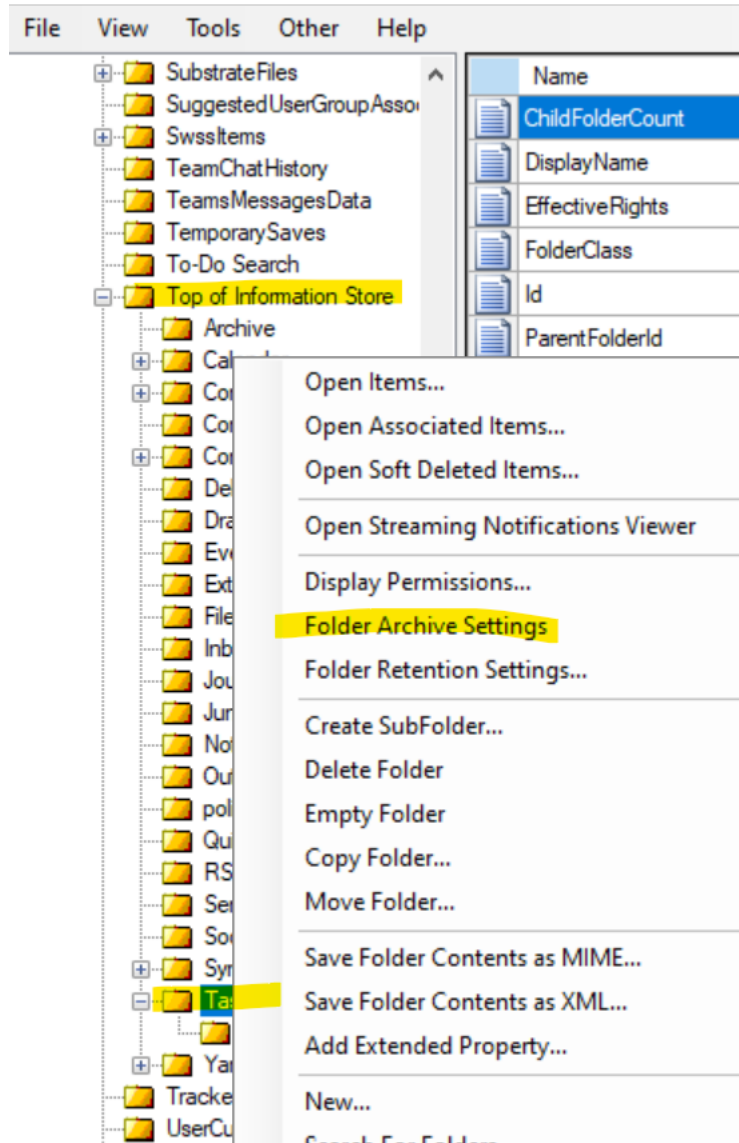
[+ Add a permission](#) Grant admin consent for

API / Permissions name	Type	Description
▼ Microsoft Graph (1)		
User.Read	Delegated	Sign in and re...
▼ Office 365 Exchange Online (1)		
EWS.AccessAsUser.All	Delegated	Access mailbo...

5.Login EWS editor



Then, after you finish the authentication, please scroll down to find here:

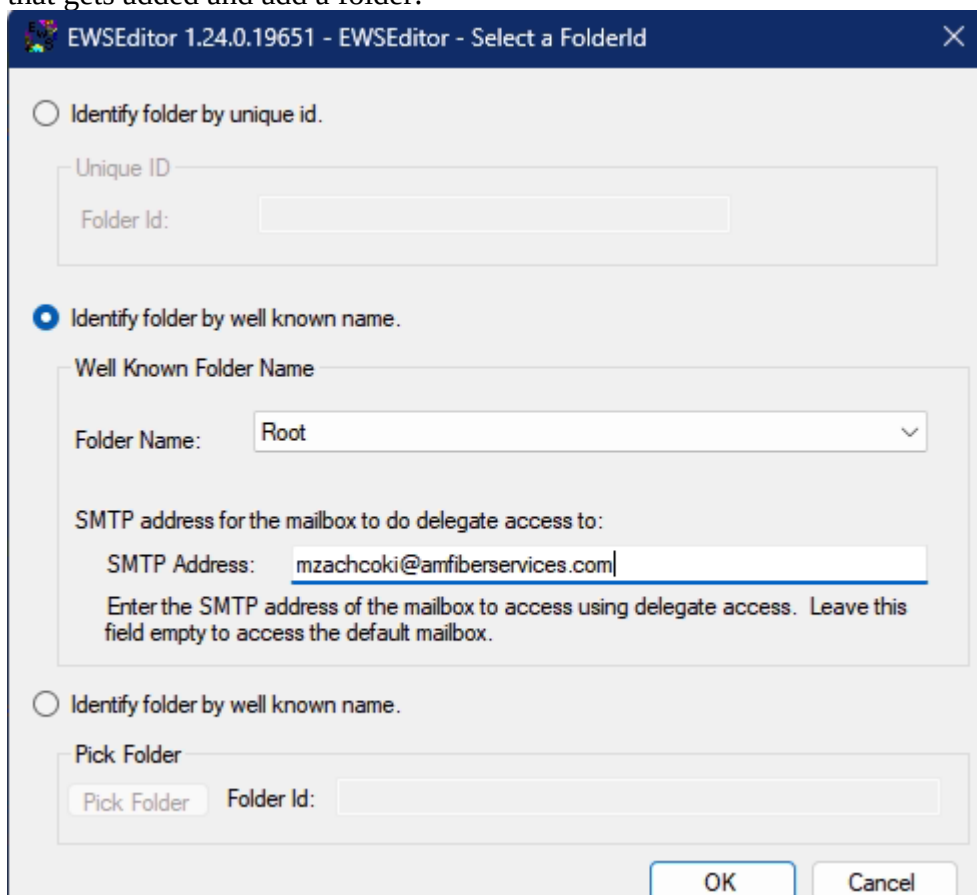


first one is tag id

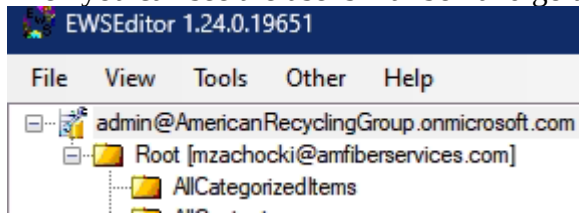


If you want to use delegation instead

if you want to use delegated permissions from the admin account rather than authenticating as the user (which could require working with them to allow MFA, etc) is to right click over the admin account that gets added and add a folder:



Then you can see the user's mailbox and go to Top Of Information Store->Tasks to apply the policies.



In addition - I found another workaround for this problem that, at least temporarily, takes care of both the archive and deletion issues, but is also less than ideal and requires some finagling. Basically, I can export the Tasks to a CSV file, then delete them, and re-import them, which creates them "as new" with a new creation date. It is a little finicky, and it doesn't work so well for recurring tasks, but then again those are the ones that aren't a problem in the first place, since when a user marks it complete, it generates a new task with a new creation date. This method does mean the archival/deletion will still occur, but now it will occur years in the future (or whenever the retention policy says).